ELSEVIER

Theoretical Computer Science 276 (2002) 445–447

Theoretical Computer Science

www.elsevier.com/locate/tcs

Note

How to decide continuity of rational functions on infinite words

Christophe Prieur *

LIAFA CNRS, Université Paris 7, 2 Place Jussieu, 75 251 Paris Cedex 05, France

Received 1 June 2000; received in revised form 8 June 2001; accepted 20 June 2001
Communicated by M. Nivat

The printed version of the paper [2] contains one mistake and two omissions of different kinds. Here are the changes needed and some explanations.

1. Correct version of the algorithm

The algorithm given in the proof of Theorem 2 is not completely correct. Let us first recall what the transducer $\overline{\mathscr{T}}$, constructed from the transducer $\mathscr{T}$, is supposed to do.

From every state q of $\mathscr{T}$ included in a path of $\mathscr{C}_2$ (resp. of $\mathscr{C}_1$) (i.e. a cyclic path labelled by the empty word on output (resp. on input)), the transducer $\overline{\mathscr{T}}$ has to simulate on input (resp. on output) all infinite iterations of paths of $\mathscr{C}_2$ (resp. of $\mathscr{C}_1$) leaving q, and on output (resp. on input), all infinite paths leaving q (under the assumption that no *sink* can be reached from q, i.e. each state reachable from q can reach an accepting state. This condition is included in the condition for $\mathscr{T}$ to be *normalized*).

As shown by the amount of "resp."-phrases in the last sentence (and in the algorithm itself), the behaviour of $\mathscr{T}$ can be divided into two separate parts: the any-path/cyclic-path simulation for states belonging to paths of $\mathscr{C}_1$, and the cyclic-path/any-path simulation for states in paths of $\mathscr{C}_2$.

Now the sets $Q_{\mathscr{C}_1}$ (the set of states belonging to a path of $\mathscr{C}_1$) and $Q_{\mathscr{C}_2}$ may have a non-empty intersection, thus the states of the transducer $\overline{\mathscr{T}}$ must keep the information of which simultation is processed. However, this information is not provided by the states of the transducer as they are defined in the paper.

* Tel.: +33-1-44-27-6133; fax: +33-1-44-27-6849.
E-mail address: christophe.prieur@liafa.jussieu.fr (C. Prieur).

PII: S0304-3975(01)00307-3

Algorithm 1 given here takes this problem into account (new states have a third component in $\{1,2\}$) and constructs the expected transducer, which means that the rest of the proof of Theorem 2 works with it.

Algorithm 1. Computation of the closure.

Data: $\mathcal{T} =< A_1, A_2, Q, q_0, F, \Delta >$
Result: $\overline{\mathcal{T}} =< A_1, A_2, Q', q_0, Q', \Delta' >$,
with $Q' \subseteq Q \cup (Q \times Q \times \{1,2\})$.
begin
$Q' \leftarrow Q$;
$\Delta' \leftarrow \Delta$;
for *each transition* $(q, a, \varepsilon, q_1) \in \mathcal{C}_2$ *(resp.* $(q, \varepsilon, b, q_2) \in \mathcal{C}_1$*)* **do**
for *each transition* (q, a', b, q_2) *(resp.* (q, a, b', q_1)*) of* $\mathcal{T}$ **do**
(a, a', b *and* b' *are either letters or the empty word.)*
$i \leftarrow 2$ (resp. $i \leftarrow 1$);
$Q' \leftarrow Q' \cup \{(q_1, q_2, i)\}$;
$\Delta' \leftarrow \Delta' \cup \{(q, a, b, (q_1, q_2, i))\}$;

Proceed similarly with new states:
for *each state* $(q_1, q_2, i) \in Q'$*, with* $i = 2$ *(resp.* $i = 1$*)* **do**
for *each transition* $(q_1, a, \varepsilon, q_1') \in \mathcal{C}_2$ *(resp.* $(q_2, \varepsilon, b, q_2') \in \mathcal{C}_1$*)* **do**
for *each transition* $(q_2, a', b, q_2') \in \Delta$ *(resp.* $(q_1, a, b', q_1') \in \Delta$*)* **do**
$Q' \leftarrow Q' \cup \{(q_1', q_2', i)\}$;
$\Delta' \leftarrow \Delta' \cup \{((q_1, q_2, i), a, b, (q_1', q_2', i))\}$;

end

2. Description of the adherence

As suggested by M. Latteux, one can actually describe more formally than above what the transducer realizing the adherence is supposed to do.

One just has to introduce the following notations, given the transducer $\mathscr{T} = \langle A_1, A_2, Q, q_0, F, \Delta \rangle$:

- $Store_1(q)$ (resp. $Store_2(q)$), with $q \in Q$, will stand for the set of finite non-empty words labelling on input (resp. on output) a cycle on q labelled on output (resp. on input) by the empty word, i.e.

$$Store_1(q) = \{u \in A_1^+ \mid \exists q \xrightarrow{u|1} q \in \Delta^+\}.$$

  This set contains all the words that can be “stored” on input (resp. on output) with no influence on the output (resp. input) word. All these words have to be considered in the computation of the adherence.
- $Next_1(q)$ (resp. $Next_2(q)$), with $q \in Q$, is the set of infinite words labelling on input (resp. output) an infinite path leaving q:

$$Next_1(q) = \{s \in A_1^\omega \mid \exists (s_i \in A_1^*)_{i \in \mathbb{N}}, (t_i \in A_1^*)_{i \in \mathbb{N}},$$

$$\exists q \overset{s_0|t_0}{\rightarrow} q_1 \overset{s_1|t_1}{\rightarrow} \cdots q_i \overset{s_i,t_i}{\rightarrow} \cdots \in \Delta^\omega$$
$$\text{and } s = s_0 s_1 \cdots s_i \cdots \}.$$

If Q' is a subset of states of the transducer $\mathscr{T}$, we will also denote by $\mathscr{T}_{Q'}$ the transducer:

$$\mathscr{T}_{Q'} = \langle A_1, A_2, Q, q_0, Q', \Delta \rangle,$$

which differs from $\mathscr{T}$ only in its terminal states.

What is proved in Theorem 2 by way of the algorithm is that, given a normalized transducer $\mathscr{T} = \langle A_1, A_2, Q, q_0, F, \Delta \rangle$, the pairs of the adherence of $||\mathscr{T}||$ that cannot be recognized by the transducer $\mathscr{T}_Q$ are of the form

$$(x \cdot u_0 u_1 \ldots u_i \ldots, \quad y \cdot t)$$
$$\text{resp.} \quad (x \cdot t, \quad y \cdot u_0 u_1 \ldots u_i \ldots),$$

where

- (x, y) is the label of a path starting in an initial state and ending in one state q,
- the words u_i are words of $Store_1(q)$ (resp. of $Store_2(q)$),
- t is the output (resp. input) label of a path leaving q, which means $t \in Next_2(q)$ (resp. $t \in Next_1(q)$).

This can be expressed by saying that the adherence of the relation realized by $\mathscr{T}$ (normalized) is the following (rational) relation:

$$\begin{aligned} Adh(\|\mathscr{T}\|_{fin}) = \|\mathscr{T}_Q\| + \bigcup_{q \in Q} \{ \|\mathscr{T}_{\{q\}}\|_{fin} \cdot (Store_1(q)^\omega \times Next_2(q) \\ + Next_1(q) \times Store_2(q)^\omega) \}. \end{aligned}$$

3. Bibliographic note

After submission of this paper, M. Nivat informed us that he had proved in [1] the fact that the topological closure of an infinitary rational relation is rational itself. The techniques used are completely different from the ones presented in our paper.

Moreover, the polynomial algorithm provided here yields a transducer realizing the closure of a relation. If the relation happens to be functional, the transducer so obtained can thus be tested to decide whether the function is continuous.

References

[1] M. Nivat, Closed infinitary rational relations, Tech. Report, Université de Picardie. Mai 1981, Journées d'Algèbre et d'Informatique en l'honneur de R. Lyndon.

[2] Ch. Prieur, How to decide continuity of rational functions on infinite words, Theoret. Comput. Sci. 250 (2001) 71–82.